What is claimed is:

1   1.   A system for providing a Java code release infrastructure with

2   granular code patching, comprising:

3   one or more Java code patches, each comprising at least one resource unit,

4   each resource unit comprising metadata and file components;

5   one or more Java code libraries, each comprising at least one such

6   resource unit;

7   a patch tool, comprising:

8   a compare module comparing the metadata for each such resource

9   unit in the Java code patches to the metadata for each such corresponding

10   resource unit in the Java code libraries; and

11   a merge module merging each such resource unit in the Java code

12   patches into the Java code libraries for each such corresponding resource unit that

13   is out-of-date.

1   2.   A system according to Claim 1, further comprising:

2   an extract module extracting at least one resource unit from the Java code

3   libraries and modifying one or more Java archive files that are out-of-date with

4   the at least one extracted resource unit.

1   3.   A system according to Claim 1, further comprising:

2   a sign module signing the Java archive files using a digital certificate.

1   4.   A system according to Claim 1, wherein the one or more Java

2   archive files are modified through at least one of creation, revision or deletion.

1   5.   A system according to Claim 1, further comprising:

2   a source repository storing the source file components;

3   a staged patch repository storing the one or more Java code patches; and

4   a staged code repository organizing the one or more Java code libraries

5   and the Java archive files.

1   6.   A system according to Claim 1, further comprising:

2   a resource unit generator processing the file components into at least one

3   such resource unit; and

4   a packager packaging at least one such resource unit into one or more of

5   the Java code patches.

1   7.   A system according to Claim 6, further comprising:

2   stored Java source code provided as the file components.

1   8.   A system according to Claim 7, further comprising:

2   a compiler compiling at least one Java source code file into one or more

3   Java classes; and

4   a resource unit packager module storing the Java classes into at least one

5   such resource unit as the file components.

1   9.   A system according to Claim 6, further comprising:

2   at least one of non-Java source and derived code provided as the file

3   components.

1   10.   A system according to Claim 6, further comprising:

2   third party code provided as the file components.

1   11.   A system according to Claim 6, further comprising:

2   a metadata generator generating the metadata for each such resource unit;

3   and

4   a resource unit packager module storing the generated metadata into the

5   resource unit.

1   12.   A system according to Claim 11, wherein the metadata comprises

2   at least one of a unique identifier and a version attribute.

1   13.   A system according to Claim 1, further comprising:

2   a compare module using a set of rules allowing one of an older resource

3   unit to be replaced by a newer resource unit and a newer resource unit to be

4   replaced by an older resource unit to back out a previously-applied Java code

5   patch.

1   14.   A system according to Claim 1, further comprising:

2   one or more Java archive files, each comprising at least one resource unit

3   corresponding to one such resource unit in the Java code libraries; and

4   a patch tool referencing Java archive file definitions which each

5   correspond to one or more of the Java archive files.

1   15.   A system according to Claim 14, further comprising:

2   an extract module extracting the resource units from the Java code

3   libraries into the Java archive files for each such corresponding resource unit that

4   is out-of-date.

1   16.   A system according to Claim 15, further comprising:

2   an extract module referencing third party Java code libraries not

3   maintained as part of the infrastructure.

1   17.   A system according to Claim 1, further comprising:

2   Java code libraries implemented as a portable virtual file system which

3   can be used directly by a Java Virtual Machine.

1   18.   A system according to Claim 1, further comprising:

2   a machine portable infrastructure providing support for Java language

3   features by encapsulating Java inner classes, nested directory structures, native

4   class names, and native character set.

1   19.   A method for providing a Java code release infrastructure with

2   granular code patching, comprising:

3   providing one or more Java code patches, each comprising at least one

4   resource unit, each resource unit comprising metadata and file components;

5   patching one or more Java code libraries, each comprising at least one

6   such resource unit;

7   comparing the metadata for each such resource unit in the Java code

8   patches to the metadata for each such corresponding resource unit in the Java

9   code libraries; and

10   merging each such resource unit in the Java code patches into the Java

11   code libraries for each such corresponding resource unit that is out-of-date.

1   20.   A method according to Claim 19, further comprising:

2   extracting at least one resource unit from the Java code libraries and

3   modifying one or more Java archive files that are out-of-date with the at least one

4   extracted resource unit.

1   21.   A method according to Claim 19, further comprising:

2   signing the Java archive files using a digital certificate.

1   22.   A method according to Claim 19, wherein the one or more Java

2   archive files are modified through at least one of creating, updating or deleting.

1   23.   A method according to Claim 19, further comprising:

2   storing source file components into a source repository;

3   storing one or more Java code patches into a staged patch repository; and

4   organizing one or more Java code libraries and the Java archive files into a

5   staged code repository.

1   24.   A method according to Claim 19, further comprising:

2   processing the file components into at least one such resource unit; and

3   packaging at least one such resource unit into one or more of the Java code

4   patches.

1   25.   A method according to Claim 24, further comprising:

2   providing Java source code as the file components.

1   26.   A method according to Claim 25, further comprising:

2   compiling at least one Java source code file into one or more Java classes;

3   and

4   storing the Java classes into at least one such resource unit as the file

5   components.

1   27.   A method according to Claim 24, further comprising:

2   providing at least one of non-Java source and derived code as the file

3   components.

1   28.   A method according to Claim 24, further comprising:

2   providing third party code as the file components.

1   29.   A method according to Claim 24, further comprising:

2   generating the metadata for each such resource unit; and

3   storing the generated metadata into the resource unit.

1   30.   A method according to Claim 29, wherein the metadata comprises

2   at least one of a unique identifier and a version attribute.

1   31.   A method according to Claim 19, further comprising:

2   using a set of rules to allow one of an older resource unit to be replaced by

3   a newer resource unit and a newer resource unit to be replaced by an older

4   resource unit to back out a previously-applied Java code patch.

1   32.   A method according to Claim 19, further comprising:

2   providing one or more Java archive files, each comprising at least one

3   resource unit corresponding to one such resource unit in the Java code libraries;

4   and

5   referencing Java archive file definitions which each correspond to one or

6   more of the Java archive files.

1   33.   A method according to Claim 32, further comprising:

2   extracting the resource units from the Java code libraries for each such

3   corresponding resource unit that is out-of-date.

1   34.   A method according to Claim 33, further comprising:

2   referencing third party Java code libraries not maintained as part of the

3   infrastructure.

1   35.   A method according to Claim 19, further comprising:

2   implementing Java code libraries as a portable virtual file system which

3   can be used directly by a Java Virtual Machine.

1   36.   A method according to Claim 19, further comprising:

2   providing a machine portable infrastructure supporting Java language

3   features by encapsulating Java inner classes, nested directory structures, native

4   class names, and native character set.

1   37.   A computer-readable storage medium holding code for performing

2   the method of Claim 19.

1   38.   A system for patching staged code in a staged Java code release

2   infrastructure, comprising:

3   a staged code repository maintaining one or more staged Java code

4   libraries, each staged Java code library comprising at least one resource unit, each

5   resource unit comprising metadata and file components;

6   a staged patch repository storing one or more Java code patches, each Java

7   code patch comprising at least one resource unit corresponding to one such

8   resource unit specified in a Java code patch definition; and

9   a patch tool accessing one or more Java code patches in the staged patch

10   repository, comprising:

11   a compare module comparing the metadata for each resource unit

12   in the Java code patches to the metadata in the staged Java code libraries for each

13   such corresponding resource unit; and

14   a merge module merging each resource unit in the Java code

15   patches into the staged Java code libraries for each such corresponding resource

16   unit that is out-of-date.

1   39.   A system according to Claim 38, further comprising:

2   an extract module referencing Java archive file definitions which each

3 correspond to a staged Java archive file, each staged Java archive file comprising

4 at least one resource unit corresponding to one such resource unit in the staged

5 Java code libraries.

1   40. A system according to Claim 39, further comprising:

2   an extract module extracting one such resource unit from the staged Java

3 code libraries into the staged Java archive files for each such corresponding

4 resource unit that is out-of-date.

1   41. A system according to Claim 40, further comprising:

2   a sign module creating a digital signature for the staged Java archive files

3 using a digital certificate.

1   42. A method for patching staged code in a Java code release

2 infrastructure, comprising:

3   maintaining one or more staged Java code libraries in a staged code

4 repository, each staged Java code library comprising at least one resource unit,

5 each resource unit comprising metadata and file components;

6   accessing one or more Java code patches in a staged patch repository, each

7 Java code patch comprising at least one resource unit corresponding to one such

8 resource unit specified in a Java code patch definition;

9   comparing the metadata for each resource unit in the Java code patches to

10 the metadata in the staged Java code libraries for each such corresponding

11 resource unit; and

12   merging each resource unit in the Java code patches into the staged Java

13 code libraries for each such corresponding resource unit that is out-of-date.

1   43. A method according to Claim 42, further comprising:

2   referencing Java archive file definitions which each correspond to a staged

3 Java archive file, each staged Java archive file comprising at least one resource

4 unit corresponding to one such resource unit in the staged Java code libraries.

1   44.   A method according to Claim 43, further comprising:

2   extracting one such resource unit from the staged Java code libraries into

3   the staged Java archive files for each such corresponding resource unit that is out-

4   of-date.

1   45.   A method according to Claim 44, further comprising:

2   creating a digital signature for the staged Java archive files using a digital

3   certificate.

1   46.   A computer-readable storage medium holding code for performing

2   the method of Claim 42.

1   47.   A system for generating Java code patches in a Java code release

2   infrastructure, comprising:

3   a source code repository maintaining one or more source files; and

4   a patch generator generating one or more Java code patches, each

5   comprising at least one resource unit, each resource unit comprising metadata and

6   file components specified in Java code patch definitions.

1   48.   A system according to Claim 47, further comprising:

2   a resource unit generator processing the file components into at least one

3   such resource unit; and

4   a packager packaging at least one resource unit into one or more of the

5   Java code patches.

1   49.   A system according to Claim 48, further comprising:

2   one or more Java source code files provided as the file components.

1   50.   A system according to Claim 49, further comprising:

2   a compiler compiling the Java source code into one or more Java classes;

3   and

4   a resource unit packager module storing the Java classes into at least one

5   such resource unit as file components.

1   51.   A system according to Claim 48, further comprising:

2   at least one of non-Java source and derived code provided as the file

3 components.

1   52.   A system according to Claim 48, further comprising:

2   staged third party code provided as the file components.

1   53.   A system according to Claim 48, further comprising:

2   a metadata generator generating the metadata for each such resource unit;

3 and

4   a resource unit packager module storing the generated metadata into one

5 such resource unit.

1   54.   A system according to Claim 53, wherein the metadata comprises

2 at least one of a unique identifier and a version attribute.

1   55.   A method for generating Java code patches in a Java code release

2 infrastructure, comprising:

3   maintaining one or more source files in a source code repository; and

4   generating one or more Java code patches, each comprising at least one

5 resource unit, each resource unit comprising metadata and file components

6 specified in Java code patch definitions.

1   56.   A method according to Claim 55, further comprising:

2   processing the file components into at least one such resource unit; and

3   packaging at least one resource unit into one or more of the Java code

4 patches.

1   57.   A method according to Claim 56, further comprising:

2   providing one or more Java source code files as the file components.

1   58.   A method according to Claim 57, further comprising:

2   compiling the Java source code into one or more Java classes; and

3   storing the Java classes into at least one such resource unit as file

4 components.

1   59.   A method according to Claim 56, further comprising:

2   providing at least one of non-Java source and derived code as the file

3 components.

1   60.   A method according to Claim 56, further comprising:

2   providing staged third party code as the file components.

1   61.   A method according to Claim 56, further comprising:

2   generating the metadata for each such resource unit; and

3   storing the generated metadata into one such resource unit.

1   62.   A method according to Claim 61, wherein the metadata comprises

2 at least one of a unique identifier and a version attribute.

1   63.   A computer-readable storage medium holding code for performing

2 the method of Claim 55.